```r
###非階層クラスター分析############################################################
library(MASS)
data(Cars93 )
str(Cars93)#data構造:型,サンプルサイズ,変数の数と内容
####欠測値処理
anyNA(Cars93[,c(5,7,12:14,17:19,25)])#指定範囲の項目にNAがあるか

###Kmeans:非階層クラスター分析の問題点:CLボリュームが初期値に依存, 最適CL数不明
##CL数探索
#NbClust関数
library(NbClust)
w.cln <- NbClust(Cars93[,c(5,7,12,14,19,22,25)],
                 min.nc = 2 , max.nc = 10,
                 diss = NULL, distance = "euclidean",
                 method = "kmeans",index="all",alphaBeale=0.1)
w.cln$Best.nc
par(mfrow=c(1,1))
barplot(table(w.cln$Best.nc[1,]))
###Within groups sum of squares:[R IN ACTION ],P.379-380から引用
d <- data.frame(scale(Cars93[,c(5,7,12,14,19,22,25)]))
cd <- scale(d)
set.seed(123)
wssplot <- function(data, nc=10 ){
    wss2 <- (nrow(data)-1)*sum(apply(data,2,var))
    for (i in 2:nc){
        wss2[i] <- sum(kmeans(data, centers=i)$withinss)}
    plot(1:nc, wss2, type = "b", pch = 17 , lwd =  2,
         xlab="Number of Clusters", ylab="Within groups sum of squares")}
wssplot(d)
wssplot(cd) #標準化data

### Kmeans ###########################################################################
set.seed(123)#seedを指定しておくと再現性がある
kms <- kmeans(x = Cars93[,c(5,7,12,14,19,22,25)] ,
              centers = 3 ,     #centers=CL数の場合重心の初期値はランダム
              iter.max = 100 ,
              #algorithm = c("Hartigan-Wong", "Lloyd",
              #              "Forgy","MacQueen"),#4種類,省略可
              nstart = 1)       #centers=CL数の場合:選択すべきrandom sets数
names(kms)
kms
##Fit
ss <- function(x) sum(scale(Cars93[,c(5,7,12,14,19,22,25)], scale = F)^2)
fitted.x <- fitted(kms);  head(fitted.x)
resid.x <- Cars93[,c(5,7,12,14,19,22,25)] - fitted(kms)
resid.x
round(kms$betweenss/kms$totss , 3)*100

##CLを決めて元dataに書き込む:kmcl$cluster
km.cl <- factor(kms$cluster)
write.csv(cbind(Cars93 , data.frame(km.cl) )
,"data/cars93.cl.csv")#CLdata書き込み

###Plot:多次元dataを2次元に縮約(Use PCA)
library(useful)
plot(data = Cars93[,c(5,7,12,14,19,22,25)] , kms )
library(cluster)
clusplot(Cars93[,c(5,7,12:14,17:19,25)],
         kms$cluster, color = T, shade = F,
         labels = 2, lines = 0)
```

```r

#####kmeans++(kmeanpp()関数)#################################################
####
#K-meansは初期重心をランダム配置するため近くに配置されることがあるがそれを避ける
#最初に1点を選択,結果を利用して重心を再配置(その後は通常のk-means)
install.packages("LICORS")
library(LICORS)
library(cluster)
library(RColorBrewer)
library(MASS)
data("Cars93")
str(Cars93)
set.seed(123)
kmpp <- kmeanspp(Cars93[,c(5,7,12,14,19,22,25)],
                 k = 3, start = "random", iter.max = 100, nstart = 1)
names(kmpp)
kmpp

##### ClusterR package ##################################################
# Gaussian mixture models, k-means, mini-batch-kmeans, k-medoids and
# affinity propagation clustering with the option to plot, validate,
# predict (new data) and estimate the optimal number of clusters.
install.packages("ClusterR")
library(ClusterR)
library(MASS)
data(Cars93)
kmrcpp <- KMeans_rcpp(Cars93[,c(5,7,12,14,19,22,25)],
                      clusters = 3 ,
                      num_init = 5 , #異なる乱数で実行される計算回数
                      max_iters = 100 , seed = 123,
                      initializer = "kmeans++")
#the method of initialization:optimal_init,quantile_init,kmeans++,random.
names(kmrcpp)
kmrcpp

#######Optimal_Clusters_KMeans()
kmopt <- Optimal_Clusters_KMeans(Cars93[,c(5,7,12,14,19,22,25)],
                                 max_clusters=10,
                                 criterion="distortion_fK",
                                 plot_clusters=T)
kmopt

###PAM K-medoids#############################################################
####kmeans:定量dataだけ,外れ値の影響大
###Kmedoids:質的dataも可,外れ値の影響小
library(cluster)
plot(pam(Cars93[,c(5,7,12,14,19,22,25,3,26)],3),ask = T)#シルエットプロット
pamx <- pam(Cars93[,c(5,7,12,14,19,22,25)] , 3)
summary(pamx)
plot(pamx)
pamx$medoids
pamx$clustering
pamx$clusinfo
pamx$silinfo
###Data merge,CLdata保存
pam.cl <- factor(pamx$cluster)
write.csv(cbind(Cars93 , data.frame(pam.cl) ) ,
          "data/cars93.pamcl.csv")


```